Econ 1034 / CS 134: Networks Problem set 9

Prof. Yaron Singer Due: **Due Wednesday, 4/12/17 at noon**

1. How to Win the Web (Easley and Kleinberg, 14.7 Q3) (20 points) In order to promote their content, designers of Web content often reason explicitly about how to create pages that will score highly on search engine rankings. This question explores some reasoning in that style.

a. **(5 points)** Show the values that you get if you run two rounds of computing hub and authority values on the network of Web pages in Figure 1 (i.e., the values computed by the k-step hub–authority computation when we choose the number of steps k to be 2). Show the values both before and after the final normalization step, in which we divide each authority score by the sum of all authority scores and divide each hub score by the sum of all hub scores. (We call the scores obtained after this dividing-down step the normalized scores. It's fine to write the normalized scores as fractions rather than decimals.)

b. **(8 points)** Now we come to the issue of creating pages to achieve large authority scores, given an existing hyperlink structure. In particular, suppose you wanted to create a new Web page X and add it to the network in Figure 1, so that it could achieve a (normalized) authority score that is as large as possible. One thing you might try is to create a second page Y as well, so that Y links to X and thus confers authority on it. In doing this, it's natural to wonder whether it helps or hurts X's authority to have Y link to other nodes as well.

Specifically, suppose you add X and Y to the network in Figure 1. To add X and Y to this network, one must specify what links they will have. Here are two options; in the first option, Y links only to X, whereas in the second option, Y links to other strong authorities in addition to X.

- Option 1: Add new nodes X and Y to Figure 1, create a single link from Y to X, and create no links out of X.
- Option 2: Add new nodes X and Y to Figure 1; create links from Y to each of A, B, and X; and create no links out of X.

For each of these two options, we'd like to know how X fares in terms of its authority score. So, for each option, show the normalized authority values that each of A, B, and X get when you run the two-step hub–authority computation on the resulting network [as in part (a)]. (That is, you should perform the normalization step in which you divide each authority value down by the total.) For which of options 1 or 2 does page X get a higher authority score (taking normalization into account)? Give a brief explanation in which you provide some intuition for why this option gives X a higher score.

c. **(7 points)** Suppose, instead of creating two pages, you create three pages, X, Y, and Z, and again try to strategically create links out of them so that X gets ranked as well as possible. Describe a strategy for adding three nodes X, Y, and Z to the network in Figure 1, with choices of links out of each, so that when you run the two-step hub–authority computation [as in parts (a) and (b)], and then rank all pages by their authority score, node X shows up in second place. [Hint: Note that there's no way to do this so that X shows up in first place, so second place is the best one can hope for using only three nodes X, Y, and Z.]

2. Limiting Values of PageRank (Easley and Kleinberg, 14.7 Q4) (20 points) Let's consider the limiting values that result from the Basic PageRank Update Rule. Recall that in the Basic PageRank Update Rule, each page divides its current PageRank score among its outgoing links equally. The new PageRank score at every node is then the sum of the values that this page receives from its incoming links. These limiting values are described as capturing "a kind of equilibrium based on direct endorsement: they are values that remain unchanged when everyone divides up their PageRank and passes it forward across their outgoing links."

This description gives a way to check whether an assignment of numbers to a set of Web pages forms an equilibrium set of PageRank values: the numbers should add up to 1, and they should remain unchanged when we apply the Basic PageRank Update Rule.

For each of the following two networks, use this approach to check whether the numbers indicated in the figure form an equilibrium set of PageRank values. (In cases where the numbers do not form an equilibrium set of PageRank values, you do not have to give numbers that do; you simply have to explain why the given numbers do not.)

a. **(10 points)** Does the assignment of numbers to the nodes in Figure 2 form an equilibrium set of PageRank values for this network of Web pages? Give an explanation for your answer.

b. **(10 points)** Does the assignment of numbers to the nodes in Figure 3 form an equilibrium set of PageRank values for this network of Web pages? Give an explanation for your answer.

3. Avoiding Undesirable Equilibria in PageRank (20 points) We consider an equilibrium set of PageRank values that result from the Basic PageRank Update Rule as in Question 2.

a. **(6 points)** What is an equilibrium set of PageRank values for the network in Figure 4?

b. **(14 points)** Consider a directed network G and assume that it is "weakly connected" (i.e. for every pair of nodes u and v in G, there is a path from u to v or a path from v to u, or both). Give a necessary and sufficient condition on the graph G for the following to be true: *All PageRank equilibiria in G give non-zero PageRank values to all nodes.* Argue precisely that this condition is both necessary and sufficient.

4. Detecting Link Farms (Easley and Kleinberg, 14.7 Q6) (20 points) One of the basic ideas behind the computation of hubs and authorities is to distinguish between pages that have multiple reinforcing endorsements and those that simply have high in-degree. Consider, for example, the graph shown in Figure 5 (Despite the fact that it has two separate pieces, keep in mind that it is a single graph). The contrast described above can be seen by comparing node D to nodes $B1$, $B2$, and $B3$: whereas D has many in-links from nodes that only point to D, nodes $B1$, $B2$, and $B3$ have fewer in-links each, but from a mutually reinforcing set of nodes. Let's explore how this contrast plays out in the context of this stylized example.

a. **(3 points)** Find the authority and hub values you get from running the 2-step hub-authority algorithm as presented in lecture on this graph (do the final normalization).

b. **(7 points)** Give formulae, in terms of k, for the authority and hub values at each node that you get from running the k-step algorithm on this graph (include the final normalization).

c. **(10 points)** As k goes to infinity, what do the normalized values at each node converge to on this graph? Give an explanation for your answer; this explanation does not have to constitute a formal proof, but it should argue at least informally why the process is converging to the values you claim, given your formulae in (b). In addition to your explanation of what's happening in the computation, briefly discuss (in 1-2 sentences) how this relates to the intuition suggested in the opening paragraph of this problem, about the difference between pages that have multiple reinforcing endorsements and those that simply have high in-degree.

5. Implementing PageRank (20 points) Sick and tired of getting rejected from job interviews (seriously: how many ways do I need to search a graph?), Raynor's decided that if he can't join Google, then he'll have to become Google. Help him in his quest for major moola by "borrowing" the PageRank (PR) algorithm and implementing it.

a. **(0 points)** We can't become Google without spying on the internet, so first we need to actually compile all the pages of the internet. **Optionally** look at the section tutorial "Web Crawling with Python" to see how to use python to index a website or group of websites. This is just for fun, but it may help you understand how Google actually...Googles.

b. **(1 points)** That said, in coding it's usually best to take the easy way out, so we're going to practice on an already compiled dataset. Load `google.txt`, a filtered list of a comprehensive crawl of actual google webpages, into a directed graph. Each node represents a webpage, and an edge from u to p signifies that page u has a link to page p. How many nodes are there? What is the average (out)-degree?

c. **(5 points)** Implement a function `pageRankIter(g,d)` that takes a graph g and a dictionary d specifying the current PR score of all the nodes in g and returns a new dictionary d_new giving the PR score of all nodes after applying one round of the basic PR update. Set g to be your google graph and d to the typical starting scores specified by the basic PR algorithm and supply a histogram of the PR scores upon return by `pageRankIter`. **For this and all later graphs: use a sensible range for your graph! A single spiky bar is not good; you don't have to include all points if it impedes the view. Make sure to include the graph(s) in your writeup!** Based on this histogram, what previous network model do you think these google pages are an example of?

d. **(6 points)** Implement a function `basicPR(g,d,k)` that takes the same definitions of g and d described previously and a number of iterations k and returns a dictionary d_new giving the PR score of all nodes after running the basic PR algorithm after k iterations. Let g and d be as described previously, and supply histograms of the PR scores after running `basicPR` for $k = 10, 50, 200$.

e. **(6 points)** Implement a function `scaledPR(g,d,k,s)` that takes the same definitions of g, d, and k described previously and a scaling factor s and returns a dictionary d_new giving the PR score of all nodes after running the **scaled** PR algorithm after k iterations. Let g and d be as described previously, $s = 0.85$, and supply histograms of the PR scores after running `scaledPR` for $k = 10, 50, 200$.

f. **(2 points)** Load the file `links.txt`, a space separated file that specifies a (fake) "link" for every node, where the first column is the node number from `google.txt` and the second column is its link. Assume RaynorSearch (TM) uses scaled Page Rank, and only returns a link as a match for a search s if s appears in the text of the link. (So `www.harvardcs134.com` is a match for ardcs, but not for 134.org). What are the top five results after RaynorSearching for 34 with $s = 0.85, k = 100$?

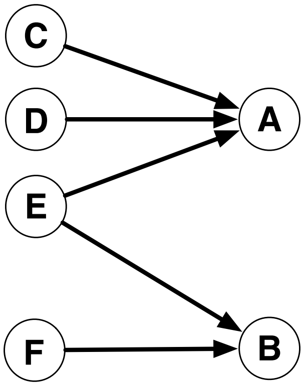

Figure 1: A network of Web pages.

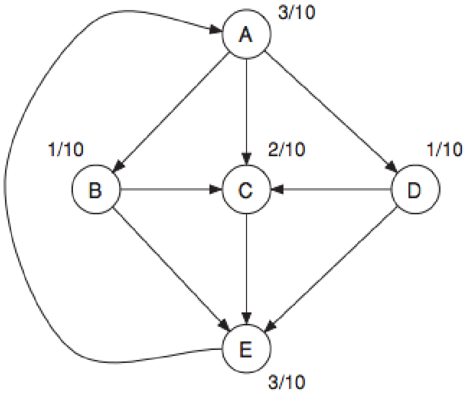

Figure 2: A network of Web pages.

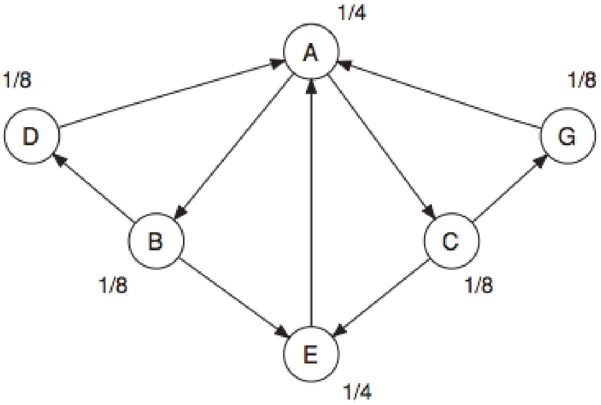

Figure 3: A network of Web pages.

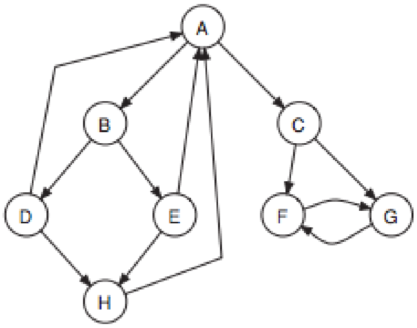

Figure 4: A network of Web pages.

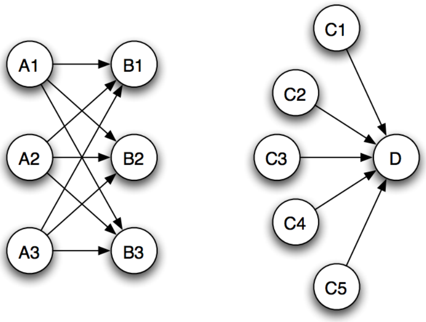

Figure 5: A network of Web pages.